LISTING OF CLAIMS

1. (Currently Amended) A computer based test bench generator for ~~verifying~~ an integrated circuit memory model, comprising:

a repository storing an identification of memory models catalogued according to memory type, number of ports and synchronous/asynchronous functional operation along with a test type associated with each such memory model;

means for entering behavior data of a memory model under test, the behavior data comprising an identification of ports in the memory model under test and a description for each such port of port cycles and port behavior;

means for entering configuration data of the memory model under test;

means for automatically generating test benches, said means being configured to select the test type from the repository ~~make a selection of suitable types of tests~~ based on a match between the entered configuration data of the memory model under test and the catalogued memory models from said repository and further execute ~~setup by executing~~ a software-based test case file generation algorithm which generates, according to the entered configuration and behavior data, specific test vectors for each of the selected test types for uniquely testing ~~that are unique to~~ the memory model under test ~~according to the configuration and behavior data~~.

2. (Canceled).

3. (Currently Amended) The test bench generator of claim 1, wherein said test benches are specified in a ~~said~~ Hardware Description Language.

4. (Previously Presented) The test bench generator of claim 3, wherein said Hardware Description Language is VERILOG.

5. (Previously Presented) The test bench generator of claim 1, wherein said behavior data is specified in a proprietary language.

6. (Previously Presented) The test bench generator of claim 1, wherein said configuration data is input to said means for generating through a command line.

7. (Previously Presented) The test bench generator of claim 1, wherein said selection of tests is based on conditional statements.

8. (Currently Amended) A method for verifying integrated circuits specified by integrated circuit memory models, comprising the steps of:

storing an identification of memory models catalogued according to memory type, number of ports and synchronous/asynchronous functional operation in a repository along with a test type associated with each memory model;

entering behavior data of a memory model under test, the behavior data comprising an identification of ports in the memory model under test and a description for each such port of port cycles and port behavior;

entering configuration data of the memory model under test;

selecting the test type ~~of suitable types of tests~~ based on a match between the entered configuration data of the memory model under test and the catalogued memory models from said repository; ~~and setting up by~~

executing a software-based test case file generation algorithm which generates, according to the configuration and behavior data, specific test vectors for each of the selected test types for uniquely testing ~~that are unique to~~ the memory model under test ~~according to the configuration and behavior data~~, so as to generate test benches; and

applying the generated test benches on a simulator to verify the memory model under test.

9. (Canceled).

10. (Currently Amended) The method according to claim 8, wherein said test benches are specified in a ~~said~~ Hardware Description Language.

11. (Previously Presented) The method according to claim 8, wherein said Hardware Description Language is VERILOG.

12. (Previously Presented) The method according to claim 8, wherein said behavior data is specified in a proprietary language.

13. (Previously Presented) The method according to claim 8, wherein said configuration data is input through a command line.

14. (Previously Presented) The method according to claim 8, wherein said selection of tests is based on conditional statements.

15. (Currently Amended) A test bench generator for integrated circuit designs, comprising:

a repository which stores functional and structural characteristic data for integrated circuit models along with a test associated with each circuit model;

a processing functionality which receives an identification of a specific integrated circuit model to be tested along with model data describing the configuration and behavior of that specific integrated circuit model, the processing functionality operating to:

process the model data in view of the identified specific integrated circuit model to produce a configured integrated circuit model suitable for simulation;

compare the specific integrated circuit model to characteristic data in the repository to identify tests applicable to that specific integrated circuit model; and

execute a software-based test case file generation algorithm which generates, in accordance with received model data, specific test vectors for each of the identified applicable tests for uniquely testing ~~that are unique to~~ the configured integrated circuit model ~~in accordance with received model data~~.

16. (Previously Presented) The generator of claim 15 wherein the specific test vectors comprise a set of self-checking test benches for the specific integrated circuit model.

17. (Original) The generator of claim 16 wherein the self-checking test benches are Verilog test benches.

18. (Original) The generator of claim 16 wherein the self-checking test benches include self-checking models incorporating complex constructs for comparing data, waiting for internal events, and timing constraint checking with respect to the specific integrated circuit model.

19. (Currently Amended) The generator of claim 15 wherein the ~~integrated circuit models in the repository, as well as the~~ received specific integrated circuit model to be tested is ~~, are~~ specified using a hardware description language.

20. (Original) The generator of claim 19 wherein the hardware description language is a Verilog language.

21. (Original) The generator of claim 15 further including a simulator functionality which applies the identified applicable tests against the configured integrated circuit model.

22. (Currently Amended) A test bench generation method for integrated circuit designs, comprising:

storing functional and structural characteristic data for integrated circuit models along with a test associated with each circuit model;

receiving an identification of a specific integrated circuit model to be tested along with model data describing the configuration and behavior of that specific integrated circuit model;

processing the model data in view of the identified specific integrated circuit model to produce a configured integrated circuit model suitable for simulation;

comparing the specific integrated circuit model to stored characteristic data ~~in the repository~~ to identify tests applicable to that specific integrated circuit model; and

executing a software-based test case file generation algorithm which generates, in accordance with received model data, specific test vectors for each of the identified applicable tests for uniquely testing ~~that are unique to~~ the configured integrated circuit model ~~in accordance with received model data~~.

23. (Previously Presented) The method of claim 22 wherein the test vectors are a set of self-checking test benches for the specific integrated circuit model.

24. (Original) The method of claim 23 wherein the self-checking test benches are Verilog test benches.

25. (Original) The method of claim 23 wherein the self-checking test benches include self-checking models incorporating complex constructs for comparing data, waiting for internal events, and timing constraint checking with respect to the specific integrated circuit model.

26. (Currently Amended) The method of claim 22 wherein ~~the integrated circuit models in the repository, as well as~~ the received specific integrated circuit model to be tested is ; ~~are~~ specified using a hardware description language.

27. (Original) The method of claim 26 wherein the hardware description language is a Verilog language.

28. (Original) The method of claim 22 further including applying the identified applicable tests against the configured integrated circuit model to simulate operation.